Linear Algebra and its Applications 436 (2012) 3215–3224

Contents lists available at SciVerse ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

QTT-rank-one vectors with QTT-rank-one and full-rank Fourier images ☆

Dmitry Savostyanov

Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina 8, Moscow 119333, Russia

ARTICLE INFO

Article history:
Received 18 July 2011
Accepted 2 November 2011
Available online 7 December 2011

Submitted by S. Friedland

AMS classification:
15A23
15A69
65F99
65T50

Keywords:
Fast Fourier transform
Quantum Fourier transform
Quantics tensor train
Data-sparse formats

ABSTRACT

Quantics tensor train (QTT), a new data-sparse format for one- and multi-dimensional vectors, is based on a bit representation of mode indices followed by a separation of variables. A radix-2 recursion, that lays behind the famous FFT algorithm, can be efficiently applied to vectors in the QTT format. If input and all intermediate vectors of the FFT algorithm have moderate QTT ranks, the resulted QTT–FFT algorithm outperforms the FFT for large vectors and has asymptotically the same complexity as the superfast quantum Fourier transform. It is instructive to describe a class of such vectors explicitly. We identify all vectors that have QTT ranks one on input, intermediate steps and output of the FFT algorithm. We also give an example of QTT-rank-one vector that has the Fourier image with full QTT ranks. We show by numerical experiments that for certain rank-one vectors with full-rank Fourier images, the practical ε-ranks remain moderate for large mode sizes.

1. Introduction

Storage for multi-dimensional arrays and complexity of algorithms working with such data grow prohibitively with the dimension. Structured low-parametric formats are necessary to make the computations feasible for large dimension. Recently, a *tensor train* (TT) format was proposed [1,2], which combines the good properties of the canonical [3–5] and Tucker [6] formats: the number of representation parameters does not grow exponentially with the dimension (there is no "*curse of dimensionality*"), and the approximation problem is stable and can be solved by the SVD-based algorithm. Surprisingly,

☆ Supported by RFBR Grants 09-01-12058, 10-01-00757, 11-01-00549, RFBR/DFG Grant 09-01-91332, Russian Federation Government Contracts No. П1112, П940, 14.740.11.0345 and Priority Research Program of Dep. Math. RAS No. 3 and 5. Part of this work was done during the stay of the author in Max-Plank Institute for Mathematics in Sciences, Inselstraße 22, 04103 Leipzig, Germany and was supported by Promotionstipendium of Max-Planck Gesellschaft.

E-mail address: dmitry.savostyanov@gmail.com

0024-3795/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2011.11.008

this format can be applied also to the low dimensional data using the *virtual levels* [7]/*quantization* of indices [8]. Quantization of a vector $x = [x(k)]_{k=0}^{n-1}$ of mode size $n = 2^d$ is the following one-to-one mapping

$$k \leftrightarrow (k_1, \dots k_d), \quad k_p = 0, 1, \quad p = 1, \dots, d, \quad k = \overline{k_1 \dots k_d} \stackrel{\text{def}}{=} \sum_{p=1}^{d} k_p 2^{p-1}, \tag{1}$$

that allows to *reshape* a vector into a *tensor* $\mathbf{X} = [x(k_1, \dots, k_d)]$ with d binary indices. The TT format for the latter is called *QTT format* and reads

$$x(k) = x(\overline{k_1 k_2 \dots k_d}) = X^{(1)}_{k_1} X^{(2)}_{k_2} \dots X^{(d)}_{k_d}, \tag{2}$$

where each $X^{(p)}_{j_p}$ is an $r_{p-1} \times r_p$ matrix and *border conditions* $r_0 = r_d = 1$ are introduced to make the right-hand side a scalar for each $k = \overline{k_1 \dots k_d}$.

The values r_p are referred to as *QTT ranks* and affect the storage and complexity in numerical work with vectors in the QTT format. As shown in [1,9,2], QTT ranks are equal to the ranks of certain *matricisations*, i.e.

$$\begin{aligned} &r_p = \operatorname{rank} X^{\{p\}}, \quad X^{\{p\}} = [x^{\{p\}}(a, b)], \\ &x^{\{p\}}(\overline{k_1 \dots k_p}, \overline{k_{p+1} \dots k_d}) = x(\overline{k_1 \dots k_d}) = x(k). \end{aligned} \tag{3}$$

Since $X^{\{p\}}$ is a $2^p \times 2^{d-p}$ matrix composed of the elements of x, the rank is bounded by row and column sizes, $r_p \leqslant \min(2^p, 2^{d-p})$. In the following we will call by "ranks of a vector" the QTT ranks of the corresponding QTT decomposition.

Definition 1. Vectors with $r_p = 1$ are referred to as *rank-one* vectors, and vectors with $r_p = 2^{\min(p, d-p)}$ as *full-rank* vectors.

Any matricisation of a random vector is a random matrix which is nonsingular with probability one (see [10]). Therefore a random vector generally (with probability one) has full QTT ranks. However, many function-related vectors have low ranks ($\exp x$, $\sin x$, $\cos x$, x^p) or have low ε-ranks, i.e. can be accurately approximated by a low-rank vectors (x^α, $e^{-\alpha x^2}$, $\frac{\sin x}{x}$, $\frac{1}{x}$, etc.) [8,11,12].

For $n = 2^d$, the normalized discrete Fourier transform (DFT) reads

$$y(j) = \frac{1}{2^{d/2}} \sum_{k=0}^{2^d-1} x(k) \omega_d^{jk}, \quad \omega_d = \exp\left(-\frac{2\pi \mathrm{i}}{2^d}\right), \quad \mathrm{i}^2 = -1, \tag{4}$$

where $F_d = \frac{1}{2^{d/2}} [\omega_d^{jk}]_{j,k=0}^{2^d-1}$ is the unitary Fourier matrix. Recently, the Fourier transform algorithm was proposed for vectors of type (2), maintaining the QTT format during the computation [13]. The complexity of m-dimensional Fourier transform of an $n \times n \times \dots \times n$ array with $n = 2^d$ is $\mathcal{O}(m^2 d^2 r^3)$, which grows logarithmically with n. For large m and n this algorithm is faster than the Fast Fourier transform (FFT) algorithm of $\mathcal{O}(mn^m \log n)$ complexity. However, it is important that r, which is the maximum QTT rank of input, output and all intermediate vectors of the algorithm, remains moderate. It is not easy to describe a class of such vectors explicitly. However, it is instructive to do this for rank-one vectors, which is the simplest case.

In this paper we describe the class of rank-one vectors with rank-one Fourier images. Also we give an example of rank-one vector that has full-rank Fourier image. This shows that Fourier transform is nontrivial operation that can increase QTT ranks of a vector to the maximum. Finally, by numerical experiments we show that practical ε-ranks of Fourier images of certain rank-one vectors (including the randomly distributed vectors) are moderate even for vectors of very large mode sizes.

2. Rank-one vectors with rank-one Fourier images

Since the QTT ranks do not change with vector scaling, we can consider only normalized vectors (zero vector, a trivial answer, is not interesting). We start from three examples of rank-one vectors that have rank-one Fourier images.

Example 1. A column of $2^d \times 2^d$ identity matrix. A unit vector

$$x = e_{k^*}, \quad \text{i.e.} \quad x(k) = \delta(k - k^*), \quad \text{where} \quad \delta(p) \stackrel{\text{def}}{=} \begin{cases} 1, & p = 0, \\ 0, & p \neq 0, \end{cases}$$

has QTT ranks one, i.e. has the decomposition (2) with all scalar cores,

$$x(k) = \delta(k - k^*) = \delta(\overline{k_1 \dots k_d} - \overline{k_1^* \dots k_d^*}) = \delta(k_1 - k_1^*) \dots \delta(k_d - k_d^*).$$

The Fourier image, $y = F_d x$, is a discretized exponent function with QTT ranks one,

$$\begin{aligned} y(j) &= \frac{1}{2^{d/2}} \exp\left(-\frac{2\pi \mathrm{i}}{2^d} k^* j\right) \\ &= \frac{1}{2^{d/2}} \exp\left(-\frac{2\pi \mathrm{i}}{2^d} k^* j_1\right) \exp\left(-\frac{2\pi \mathrm{i}}{2^{d-1}} k^* j_2\right) \dots \exp\left(-\frac{2\pi \mathrm{i}}{2} k^* j_d\right). \end{aligned}$$

Example 2. A vector which is a discretized exponent function,

$$x = [x(k)], \quad x(k) = \frac{1}{2^{d/2}} \exp\left(\frac{2\pi \mathrm{i}}{2^d} fk\right), \quad k = 0, \dots, 2^d - 1, \tag{5}$$

with integer $f = j^*$, has all QTT ranks one as well as its Fourier image $y = F_d x = e_{j^*}$.

Example 3. For $d = 1$ any vector of size $2^d = 2$ has both QTT ranks one, $r_0 = r_1 = 1$, as well as its Fourier image.

In the following we will show that any rank-one vector with rank-one Fourier image can be represented as a tensor product of the considered examples. Note that the QTT cores of a rank-one vector are unique up to the scaling, i.e. the scaling coefficient can be arbitrary distributed between the QTT cores without changing the result. In the following theorem we describe a class of equivalent rank-one QTT decompositions by one element, and show that a particular QTT representation of the rank-one vector with rank-one Fourier has a specific form.

Theorem 1. *A rank-one vector x of size 2^d has rank-one Fourier image, iff the QTT decomposition*

$$x(k) = x(\overline{k_1 k_2 \dots k_d}) = x_{k_1}^{(1)} x_{k_2}^{(2)} \dots x_{k_d}^{(d)},$$

after appropriate scaling of QTT cores can be written for some $c = 1, \dots, d$ as follows:

$$\begin{aligned} &x_{k_p}^{(p)} = \delta(k_p - k_p^*), \quad k_p^* = 0 \ \text{or} \ 1, \quad \text{for} \ \ p < c; \\ &x_{k_c}^{(c)} \ \ \text{is arbitrary;} \\ &x_{k_p}^{(p)} = \frac{1}{\sqrt{2}} \exp\left(\frac{2\pi i}{2^{d-p+1}} j^* k_p\right), \quad \text{for} \ \ p > c. \end{aligned} \tag{6}$$

Proof. For $d = 1$ the statement holds, see Example 3. Suppose it holds for any vector of size 2^{d-1} and consider the rank-one vector with rank-one Fourier image of size 2^d,

$$y_{j_1}^{(1)} y_{j_2}^{(2)} \dots y_{j_d}^{(d)} = \frac{1}{2^{d/2}} \sum_{k_1 \dots k_d} x_{k_1}^{(1)} \dots x_{k_{d-1}}^{(d-1)} x_{k_d}^{(d)} \omega_d^{jk}.$$

Write these equations separately for $j_1 = 0$ and $j_1 = 1$,

$$\begin{aligned} y_0^{(1)} y_{j_2}^{(2)} \ldots y_{j_d}^{(d)} &= \frac{x_0^{(d)} + x_1^{(d)}}{2^{d/2}} \sum_{k_1 \ldots k_{d-1}} x_{k_1}^{(1)} \ldots x_{k_{d-1}}^{(d-1)} \quad \omega_{d-1}^{j'k'}, \\ y_1^{(1)} y_{j_2}^{(2)} \ldots y_{j_d}^{(d)} &= \frac{x_0^{(d)} - x_1^{(d)}}{2^{d/2}} \sum_{k_1 \ldots k_{d-1}} x_{k_1}^{(1)} \ldots x_{k_{d-1}}^{(d-1)} \omega_d^{k'} \omega_{d-1}^{j'k'}, \end{aligned} \tag{7}$$

where $k = \overline{k_1 \ldots k_d}$, $j' = \overline{j_2 \ldots j_d}$, $k' = \overline{k_1 \ldots k_{d-1}}$. We come to the radix-2 recursive relation, that was known to Gauss [14,15] and lays behind the Cooley–Tukey FFT algorithm [16]. If both $y_0^{(1)} = 0$ and $y_1^{(1)} = 0$ then $y = F_d x = 0$ and since F_d is nonsingular we have $x = 0$, a trivial case. Three non-trivial cases are possible.

First, $y_0^{(1)} \neq 0$ and $y_1^{(1)} = 0$, leads to $x_0^{(d)} = x_1^{(d)}$ and $y' = F_{d-1} x'$, where half-size vectors x' and y' have QTT ranks one,

$$x'(k') = x'(\overline{k_1 \ldots k_{d-1}}) = x_{k_1}^{(1)} \ldots x_{k_{d-1}}^{(d-1)}, \quad y'(j') = y'(\overline{j_2 \ldots j_d}) = y_{j_2}^{(2)} \ldots y_{j_d}^{(d)}.$$

Second case, $y_0^{(1)} = 0$ and $y_1^{(1)} \neq 0$, leads to $x_0^{(d)} = -x_1^{(d)}$ and $y' = F_{d-1} \Omega_d x'$, where $\Omega_d = \operatorname{diag}\{\omega_d^{k'}\}_{k'=0}^{2^{d-1}-1}$. Note that $\Omega_d x'$ has QTT ranks one as well as x'. With proper scaling, we summarize these two cases to

$$y_{j_1}^{(1)} = \delta(j_1 - j_1^*), \quad x_{k_d}^{(d)} = \frac{1}{\sqrt{2}} \exp\left(\frac{2\pi \mathrm{i}}{2} j_1^* k_d\right), \quad y' = F_{d-1} \Omega_d^{j_1^*} x',$$

where $j_1^* = 0, 1$ and the vectors y' and $\Omega_d^{j_1^*} x'$ have size 2^{d-1} and QTT ranks one. By the assumption, cores of the vector $\Omega_d^{j_1^*} x'$ are given by (6), which means

$$\begin{aligned} &\exp\left(-\frac{2\pi \mathrm{i}}{2^{d-p+1}} j_1^* k_p\right) x_{k_p}^{(p)} = \delta(k_p - k_p^*), \quad k_p^* = 0 \ \text{ or } \ 1, \quad p = 1, \ldots, c-1; \\ &\exp\left(-\frac{2\pi \mathrm{i}}{2^{d-c+1}} j_1^* k_c\right) x_{k_c}^{(c)} \quad \text{is arbitrary;} \\ &\exp\left(-\frac{2\pi \mathrm{i}}{2^{d-p+1}} j_1^* k_p\right) x_{k_p}^{(p)} = \frac{1}{\sqrt{2}} \exp\left(\frac{2\pi \mathrm{i}}{2^{d-p}} j'_* k_p\right), \quad p = c+1, \ldots, d-1. \end{aligned}$$

Moving the scaling coefficients to the core $x^{(c)}$, we result in Eq. (6) with $j^* = 2j'_* + j_1^*$.

Finally, consider the case $y_0^{(1)} \neq 0$ and $y_1^{(1)} \neq 0$ in (7). Then

$$y' = \alpha F_{d-1} x' = \beta F_{d-1} \Omega_d x',$$

with above-defined x' and y' and some non-zero scalars α and β, that can be always chosen unit in modulus. Since F_{d-1} is nonsingular, the last equation gives

$$x'(k') = e^{\mathrm{i}\varphi} \exp\left(-\frac{2\pi \mathrm{i}}{2^d} k'\right) x'(k'), \quad k' = 0, \ldots, 2^{d-1} - 1,$$

that holds only if x' has all but one zero elements. Therefore, with proper scaling this vector is a column of identity $2^{d-1} \times 2^{d-1}$ matrix, $x' = e_{k'_*}$. We obtain the QTT decomposition of the form (6) with $c = d$.

Since all possible cases result in Eq. (6), the proof is complete. $\square$

Directly from the proof of Theorem 1 we conclude the following.

Corollary 1. *If a rank-one vector of size 2^d has rank-one Fourier image then all intermediate vectors of the Cooley–Tukey FFT algorithm have QTT ranks one.*

3. Rank-one vector with full-rank image

In the Example 2 we show that the rank-one discrete exponential function (5) has rank-one Fourier image for integer $f = j^*$. Now we consider $f \notin \mathbb{Z}$ and prove that Fourier image $y = F_d x$ has full QTT ranks. Using the power series formula, we compute

$$y(j) = \frac{1}{2^d} \sum_{k=0}^{2^d-1} \exp\left(\frac{2\pi\mathrm{i}}{2^d} fk\right) \exp\left(-\frac{2\pi\mathrm{i}}{2^d} jk\right) = \frac{1}{2^d} \frac{1-\exp\left(2\pi\mathrm{i}(f-j)\right)}{1-\exp\left(\frac{2\pi\mathrm{i}}{2^d}(f-j)\right)}.$$

Then, using $1 - e^{2\mathrm{i}\varphi} = -2\mathrm{i}e^{\mathrm{i}\varphi}\sin\varphi$, we come to

$$y(j) = \frac{1}{2^d} \frac{\exp(\pi\mathrm{i}f)\sin(\pi f)}{\exp\left(\frac{\pi\mathrm{i}}{2^d}(f-j)\right)\sin\left(\frac{\pi}{2^d}(f-j)\right)} = \alpha \frac{\exp\left(\frac{2\pi\mathrm{i}}{2^d} j\right)}{\sin\left(\frac{\pi}{2^d}(j-f)\right)},$$

where $\alpha = \frac{1}{2^d}\exp\left(\pi\mathrm{i}f\left(1-\frac{1}{2^d}\right)\right)\sin\pi f$. By (3), QTT ranks of Fourier image y are equal to the ranks of matricisations,

$$r_p = \operatorname{rank} Y^{\{p\}}, \quad Y^{\{p\}} = \left[y^{\{p\}}(a,b)\right], \quad a = \overline{j_1 \ldots j_p}, \quad b = \overline{j_{p+1}\ldots j_d},$$

$$y^{\{p\}}(a,b) = y(a+2^p b) = \alpha \frac{\exp\left(\frac{2\pi\mathrm{i}}{2^d}a + \frac{2\pi\mathrm{i}}{2^q}b\right)}{\sin\left(\frac{\pi}{2^d}(a-f) + \frac{\pi}{2^q}b\right)},$$

where $p + q = d$. Also, $f = 2^p g + h + \varphi$, where $g \in \mathbb{Z}$, $h = 0, \ldots, 2^p - 1$ and $0 < \varphi < 1$. Finally, we represent $\varphi = \varphi_1 + 2^p\varphi_2$ and write

$$\sin\left(\frac{\pi}{2^d}(a-f) + \frac{\pi}{2^q}b\right) = \sin\frac{\pi}{2^d}a' \cos\frac{\pi}{2^q}b' + \cos\frac{\pi}{2^d}a' \sin\frac{\pi}{2^q}b',$$

resulting in $Y^{\{p\}} = \alpha\, ACB$ with

$$A = \operatorname{diag}\left\{\frac{\exp\frac{2\pi\mathrm{i}}{2^d}a}{\sin\frac{\pi}{2^d}a'}\right\}, \quad C = \left[\frac{1}{\cot\frac{\pi}{2^d}a' + \cot\frac{\pi}{2^q}b'}\right], \quad B = \operatorname{diag}\left\{\frac{\exp\frac{2\pi\mathrm{i}}{2^q}b}{\sin\frac{\pi}{2^q}b'}\right\}, \tag{8}$$

where $a = 0, \ldots, 2^p - 1$, $b = 0, \ldots, 2^q - 1$ and $a' = a - h - \varphi_1$, $b' = b - g - \varphi_2$. This representation is correctly defined if all denominators in (8) are non-zero,

$$\begin{array}{l} \sin\frac{\pi}{2^d}(a-h-\varphi_1) \neq 0; \\ \cot\frac{\pi}{2^d}(a-h-\varphi_1) + \cot\frac{\pi}{2^q}(b-g-\varphi_2) \neq 0; \quad \text{for} \quad \begin{array}{l} a = 0, \ldots, 2^p-1; \\ b = 0, \ldots, 2^q - 1, \end{array} \\ \sin\frac{\pi}{2^q}(b-g-\varphi_2) \neq 0; \end{array}$$

that always can be achieved by proper choice of φ_1 and φ_2. The diagonal matrices A and B are non-singular since all diagonal elements are non-zero. The rectangular $2^p \times 2^q$ matrix C contains a square submatrix $\left[\frac{1}{s_a+t_b}\right]$ with $a, b = 0, \ldots 2^{\min(p,q)} - 1$, that is also non-singular [17], since it is a Cauchy-Hilbert matrix with distinct s_a and t_b. We conclude that matricisation $Y^{\{p\}}$ has full rank, $r_p = 2^{\min(p,d-p)}$ and therefore vector y has full QTT ranks.

4. Numerical experiments

Since the accuracy of numerical computations is limited (at least due to the machine precision round-off errors), tensor decompositions in scientific computing often do not represent a given tensor exactly. Therefore, instead of tensor decompositions we consider tensor approximations and the corresponding ε-ranks,

$$r_\varepsilon(z) = \min_{\tilde{z}:\|z-\tilde{z}\|\leqslant\varepsilon\|z\|} r(\tilde{z}). \tag{9}$$

Since the QTT format has d possibly different ranks, it is more convenient to introduce one value $r(\tilde{z})$ to describe the number of parameters that are used to represent $\tilde{z}$ in QTT format. The maximum QTT rank can be used, but sometimes it gives incorrect impression of the "structure complexity". To account the distribution of all TT ranks $r_1, \ldots, r_{d-1}$, that affect the storage for TT format, we define the *effective* TT rank.

Definition 2. [1] The effective QTT rank r of the TT format with TT ranks $r_1, \ldots, r_{d-1}$ and mode sizes $n_1, \ldots, n_d$, is as a positive solution of the quadratic equation

$$\mathrm{mem}(r_1, \ldots, r_{d-1}) = \mathrm{mem}(r, \ldots, r),$$

where $\mathrm{mem}(r_1, \ldots, r_{d-1})$ denotes the amount of memory to store the TT cores,

$$\mathrm{mem}(r_1, \ldots, r_{d-1}) = n_1 r_1 + r_1 n_2 r_2 + \cdots + r_{d-1} n_d.$$

For the QTT format, all mode sizes are equal 2 and effective QTT rank is the positive solution of quadratic equation

$$(d-2)r^2 + 2r - \sum_{p=1}^{d} r_{p-1} r_p = 0. \tag{10}$$

The effective rank is generally a non-integer value. The effective rank of rank-one vector is equal to one. The effective rank of full-rank vector of size 2^d is $r_{\text{full}} \approx \sqrt{2^d/d}$ and grows exponentially with d, as well as the storage of full array.

In Fig. 1(left) we show the effective ε-rank of the Fourier image of the discretized exponential function $x = \left[\exp\left(\frac{2\pi \mathrm{i}}{2^d} fk\right)\right]_{k=0}^{2^d-1}$ with frequency $0 \leqslant f \leqslant 1$ for different accuracy levels ε. We see that $r_\varepsilon(f)$ tends to one in the small neighborhoods of zero and one, and almost does not depend on f at certain distance from the sides of the interval. Therefore, we can say that the ε-ranks $r_\varepsilon(f)$ of the discretized exponential functions with frequency f are sensitive to ε but not to f for the most of the values of f. Note also that ε-ranks remain moderate even when the accuracy ε is close to the machine precision, although the exact decomposition is full-rank. The effective ε-rank of the Fourier image even reduces slightly for vectors of very large size $n = 2^{60}$. This shows that even for "bad examples" of data with full-rank images, the approximate Fourier transform using the QTT–FFT algorithm may sufficiently reduce the storage and complexity in comparison with usual FFT.

It is interesting to compare this result with the distribution of ε-ranks of Fourier images of randomly chosen rank-one vectors, see Fig. 1(right). For the QTT vector (2) we set $x_0^{(p)} = 1$ for $p = 1, \ldots, d$ and take $x_1^{(p)}$ with real and imaginary parts independently and uniformly distributed in $[0 : 1]$. We use 5×10^8 samples of vectors of size $n = 2^{30}$ and 10^8 samples of vectors of size $n = 2^{60}$. It is natural to expect that Fourier images of random vectors would not have a good structure. However, we can see that effective ε-ranks are again quite moderate. Also, the histograms which estimate the probability density function of the effective ε-rank of the Fourier image of random rank-one vectors are very narrow. As in the previous example, we can say that effective ε-ranks actually depend only on accuracy ε and are almost the same for most of the random vectors from the selected set.

The program code for numerical experiments (including QTT–FFT and necessary TT subroutines) was developed in `Fortran90` by the author. The computations were done using 1024 cores of SKIF-MGU "Chebyshëv" cluster, Moscow State University, Russia.

[1] This definition was proposed by E.E. Tyrtyshnikov.

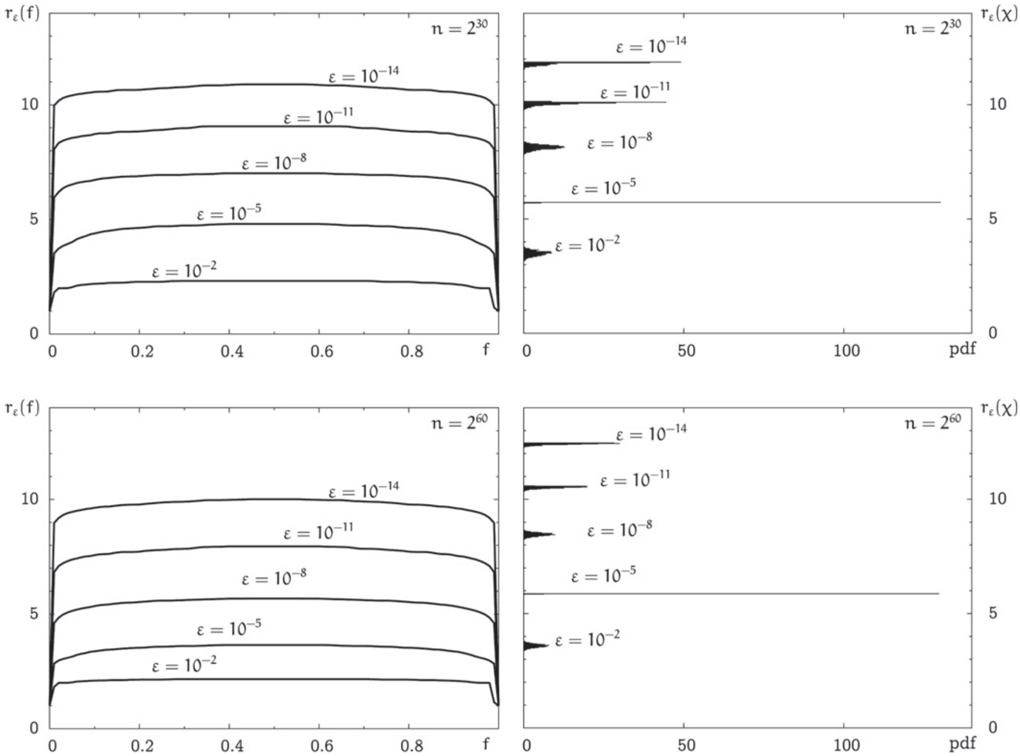

Fig. 1. (left) Effective ε-rank of the Fourier image of $x = \left[\exp\left(\frac{2\pi \mathrm{i}}{2^d} fk\right)\right]_{k=0}^{2^d-1}$, $0 \leqslant f \leqslant 1$; (right) probability density function of the effective ε-rank of the Fourier image of random rank-one vectors; (top) vectors of size $n = 2^{30}$; (bottom) vectors of size $n = 2^{60}$.

5. Discussion

In this section we discuss the links between the QTT–FFT algorithm and results from quantum computing and signal processing, which justify the importance of the addressed problem and reveal some possible directions for the further work. We also explain, how the results of this paper are generalized to the multi-dimensional case.

5.1. QTT–FFT and quantum Fourier transform

The QTT–FFT algorithm can be compared with the related approaches from quantum computing. A vector of size 2^d is commonly identified in quantum information theory with an entangled quantum state of a system of d qubits, i.e. systems with two levels. This is exactly what we do by ‘bit representation’ (1). The QTT format (2) was already known in quantum chemistry as the matrix product states (MPS) for quantum spin chains [18], but optimization techniques for MPS [19,20] are different from the algebraic SVD-based methods proposed in [9,2] and used in QTT–FFT. Probably, this close relation between QTT approach and quantum algorithms motivated the use of the letter “Q” in the abbreviation “QTT”. However, I personally think that the word “quantics” [2] is slightly misleading in this context and probably should be replaced by “qubit tensor train” to emphasize the existing links with quantum information theory and algorithms.

[2] Concise Oxford Dictionary: `quantic` *noun*: *Mathematics* a homogeneous function of two or more variables having rational or integral coefficients.

The Fourier transform is utilized in many quantum algorithms including eigenvalue estimation, order-finding and integer factorization (see [21,22] and references therein). The quantum Fourier transform (QFT) writes as a sequence of one- and two-qubit *gates*, i.e. operations involving only one or two qubits of the d-qubit system. An one-qubit gate (as well as an operating with single QTT core) modifies all 2^d entries of the entangled d-qubit state (or data vector). Due to this "exponential performance" of qubit gates the complexity of the superfast QFT algorithm is as small as $\mathcal{O}(d^2)$ quantum operations. The complexity of the FFT on the classical computer is $\mathcal{O}(2^d d)$ and grows exponentially in d. This can be considered as a manifestation of the principal difference between the quantum and classical systems noted by Richard P. Feynman in 1982. He declared that quantum mechanics cannot be efficiently simulated by classical means, i.e. a classical simulation of a *general* quantum evolution suffers from the curse of dimensionality [23].

Nevertheless, a decent *classical model* of a quantum algorithm can effectively simulate a *certain class* of quantum processes. For example, Vidal in 2003 considered the efficient classical simulation of slightly entangled quantum systems [19]. The QTT–FFT algorithm, which can be considered as a classical model of QFT, has the complexity of $\mathcal{O}(d^2 r^3)$ operations in 1D case. In general, r can grow exponentially in d, as r_{full} really does for a full-rank vector. For a certain class of vectors, however, r is bounded in respect to d by some moderate value and the QTT–FFT has asymptotically the same complexity as QFT, i.e. QTT–FFT models QFT *efficiently*. In this paper we identified all vectors with $r = 1$ and demonstrate by numerical experiments that many other vectors are close to vectors with moderate r. This gives us a hope that QTT–FFT can be used to solve important problems of quantum computing.

Note that the Hadamard transform, also very popular in quantum computing, has QTT rank one by the definition,

$$H_d = H_1^{\otimes d}, \quad H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

and does not change the QTT ranks of a vector. Therefore, a quantum Hadamard transform can be efficiently modeled by classical means for all vectors of moderate QTT rank, which are considered in [8,11,12].

Finally, it is worth noting the recent progress in the computation of discrete convolution of vectors given in the QTT format [24,25]. This operation is very important in scientific computing and can be conducted by three Fourier transforms. If QTT–FFT is applied to compute the Fourier transform, the discrete convolution of two vectors of size 2^d and QTT ranks r_1 and r_2 requires $\mathcal{O}(d^2\,\texttt{poly}(r_1, r_2))$ operations, i.e. complexity is quadratic in d and polynomial in ranks. However, in [25] an algorithm of complexity $\mathcal{O}(dr_1^2 r_2^2)$, which is linear in d, is proposed. This may indicate that in quantum computing the convolution is in some respect "more simple" operation than the Fourier transform. We can also assume that there exists an algorithm that computes the convolution in $\mathcal{O}(d)$ quantum operations (not through three QFTs). However, we did not find a reference to such algorithm in the literature.

5.2. QTT–FFT and uncertainty principle

Considering the QTT format as a data compression method we can compare it with a sparse representation popular in signal processing. The well-known *uncertainty principle* bounds the joint sparsity of a signal and its Fourier transform [26,27]: both of them cannot be very sparse. However, in terms of *data-sparsity* provided by QTT approach, we can circumvent this restriction. We say that vector is data-sparse if it can be defined in some structured form by a small number of parameters. Eq. (6) describes a class of extremely data sparse vectors (QTT-rank-one) that have extremely data-sparse Fourier images. Numerical experiments show that there are much more vectors that have data-sparse approximation together with their Fourier images.

5.3. QTT–FFT in multi-dimensional case

Finally, we should explain how the results of this paper can be generalized to the multi-dimensional Fourier transform. The QTT format for multi-dimensional data is basically the same as (2). However, it is important to choose the ordering of bit indices, that allows a low-rank representation of arrays in the computations. Most common are *straight* ordering,

$$ij \cdots k \leftrightarrow i_1 i_2 \ldots i_d \, j_1 j_2 \ldots j_d \;\cdots\; k_1 k_2 \ldots k_d,$$

and *mixed* ordering

$$ij \cdots k \leftrightarrow i_1 j_1 \cdots k_1 \; i_2 j_2 \cdots k_2 \;\ldots\; i_d j_d \cdots k_d.$$

Here letters $i, j, \ldots, k$ denote m different modes and numbers denote d bits used in each mode. The multi-dimensional Fourier transform can be implemented in QTT format with both orderings, but is especially simple in the straight ordering, since in this case it is a tensor product of 1D Fourier transforms. The QTT–FFT algorithm given in [13] assumes straight ordering for the QTT representation of multi-dimensional data. In this case the results of this paper are directly applicable to multi-dimensional vectors.

Acknowledgements

I thank the reviewers of this paper for their work and helpful comments. I am very grateful to the anonymous referee, who encourages me to discuss the existing links between QTT approach to FFT and approaches from quantum computing and signal processing.

References

[1] I.V. Oseledets, A new tensor decomposition, Dokl. Math. 80 (1) (2009) 495–496, doi:10.1134/S1064562409040115.
[2] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (5) (2011) 2295–2317, doi:10.1137/090752286.
[3] F.L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, J. Math. Phys. 6 (1) (1927) 164–189.
[4] R.A. Harshman, Foundations of the Parafac procedure: models and conditions for an explanatory multimodal factor analysis, UCLA Working Papers Phonetics 16 (1970) 1–84.
[5] J.D. Caroll, J.J. Chang, Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart–Young decomposition, Psychometrika 35 (1970) 283–319, doi:10.1007/BF02310791.
[6] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (1966) 279–311, doi:10.1007/BF02289464.
[7] E.E. Tyrtyshnikov, Tensor approximations of matrices generated by asymptotically smooth functions, Sb. Math. 194 (6) (2003) 941–954, doi:10.1070/SM2003v194n06ABEH000747.
[8] B.N. Khoromskij, $\mathcal{O}(d \log n)$-Quantics approximation of N–d tensors in high-dimensional numerical modeling, Constr. Approx. 34 (2) (2011) 257–280, doi:10.1007/s00365-011-9131-1.
[9] I.V. Oseledets, Approximation of $2^d \times 2^d$ matrices using tensor decomposition, SIAM J. Matrix Anal. Appl. 31 (4) (2010) 2130–2145, doi:10.1137/090757861.
[10] X. Feng, Z. Zhang, The rank of a random matrix, Appl. Math. Comput. 185 (3) (2007) 689–694, doi:10.1016/j.amc.2006.07.076.
[11] L. Grasedyck, Polynomial approximation in hierarchical Tucker format by vector-tensorization, 2010. Available from: < http://www.dfg-spp1324.de/download/preprints/preprint043.pdf> (DFG-SPP1324 preprint 43).
[12] I. Oseledets, Constructive representation of functions in tensor formats, 2010. Available from: < http://pub.inm.ras.ru> (preprint 2010-04).
[13] S.V. Dolgov, B.N. Khoromskij, D.V. Savostyanov, Multidimensional Fourier transform in logarithmic complexity using QTT approximation, J. Fourier Anal. Appl., submitted for publication. Available from: < http://www.mis.mpg.de/preprints/2011/preprint2011_18.pdf> (preprint 18).
[14] C.F. Gauss, Nachlass: Theoria interpolationis methodo nova tractata, in: Werke, vol. 3, Königliche Gesellschaft der Wissenschaften, Göttingem, 1866, pp. 265–330.
[15] M.T. Heideman, D.H. Johnson, C.S. Burrus, Gauss and the history of the fast Fourier transform, Arch. Hist. Exact Sci. 34 (3) (1985) 265–277, doi:10.1007/BF00348431.
[16] J.W. Cooley, J.W. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comput. 19 (1965) 297–301, doi:10.2307/2003354.
[17] S. Schechter, On the inversion of certain matrices, Math. Tables Other Aids Comput. 66 (13) (1959) 73–77.
[18] M. Fannes, B. Nachtergaele, R. Werner, Finitely correlated states on quantum spin chains, Commun. Math. Phys. 144 (3) (1992) 443–490.
[19] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, Phys. Rev. Lett. 91 (14) (2003) 147902.
[20] S.R. White, Density-matrix algorithms for quantum renormalization groups, Phys. Rev. B 48 (14) (1993) 10345–10356, doi:10.1103/PhysRevB.48.10345.
[21] A. Ekert, R. Jozsa, Quantum algorithms: entanglement-enhanced information processing, Philos. Trans. R. Soc. Lond. 356 (1998) 1769–1782.
[22] P. Kaye, R. Laflamme, M. Mosca, An Introduction to Quantum Computing, Oxford University Press, 2007.
[23] C.H. Papadimitriou, Computational Complexity, Addison Wesley, 1994.

[24] W. Hackbusch, Tensorisation of vectors and their efficient convolution, 2010. Available from: < http://www.dfg-spp1324.de/download/preprints/preprint065.pdf> (DFG-SPP1324 preprint 65).
[25] V. Kazeev, B.N. Khoromskij, E.E. Tyrtyshnikov, Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity, Tech. Rep. 36, SISC, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, submitted for publication. Available from: < http://www.mis.mpg.de/publications/preprints/2011/prepr2011-36.html>.
[26] D.L. Donoho, P.B. Stark, Uncertainty principle and signal recovery, SIAM J. Appl. Math. 49 (3) (1989) 906–931, doi:10.1137/0149053.
[27] M. Elad, A.M. Bruckstein, A generalized uncertainty principle and sparse representation in pairs of bases, IEEE Trans. Inform. Theory 48 (9) (2002) 2558–2567, doi:10.1109/TIT.2002.801410.